Detection and Prevention Methods of Botnet-generated Spam

Areej Al-Bataineh
Computer Science Department
University of Texas at San Antonio
hpp239@my.utsa.edu

Gregory White
Computer Science Department
University of Texas at San Antonio
greg@utsa.edu

Abstract

Although anti-spam measures are improving, the spam volume is increasing due to the use of *Botnets*. Botnets facilitate an efficient generation and guaranteed delivery of large volumes of spam. Spambots, or spam-generating bots, use different transmission methods based on the network settings of the infected host. These methods include relaying, proxying, and direct delivery. In this paper, we illustrate these methods and discuss what measures can be taken against them to address the spamming botnet threat. These measures are divided by the place they can be adopted at; the edge routers and the mail servers.

Keywords

Spam, Botnet.

1 Introduction

Protecting networks and hosts from being targets of attacks becomes a more challenging task with the increased sophistication of attack tools. Botnets, or networks of compromised machines, are one of these tools. They are considered one of the most dangerous threats to the network security today [1]. They are being used as a vehicle for an array of cybercrimes, such as spamming, denial of service, identity theft, and phishing.

The research community is actively looking for effective methods against this phenomenon. As a result, a number of botnet-specific mitigation tools have been produced. These tools can be applied either on the network level or on the host level. Most of these tools focus on discovering the Command and Control (C&C) channels between botnet controllers and individual bots. These tools work for some botnets, and do not work for others because botnet developers keep looking for ways to evade detection. For example, modern botnets employ encryption and custom protocols for their communications.

When botnet-mitigation tools fail, another layer of security is needed. This entails preventing bots from carrying out attacks; i.e. bot neutralization. In this paper, we focus on the number one use of botnets, which is spamming [2]. Spamming is the process of creating unsolicited bulk emails and sending them to a large number of users. Although spam is not a new problem, the use of botnets has increased its volume exponentially [3]. On the other hand, botnet usage brings out new dimensions for solutions.

In this paper, we try to answer the following questions. How do botnets transmit spam? What can be done to make it nearly impossible for botnets to deliver spam? What tools and policies can be utilized to prevent spam traffic from traversing our networks?

This paper is structured as follows. In Section 2 we provide an overview of the botnet phenomenon and in Section 3 we illustrate the structure and the operation of a general spamming botnet. Spam transmission methods are discussed with some detail in Section 3. In Section 4 we provide a survey of spam prevention methods that can be undertaken at two places within any network, the edge routers, and the mail servers. We conclude our findings in the last section.

2 Botnets

A *Botnet* is a network of compromised machines, or *Bots*, that is under the command and control (C&C) of one person, called the *master* [4]. Machines can be compromised and infected with malware in a variety of ways; such as clicking on malicious email attachments, visiting malicious websites, and installing software from untrusted sources. It can also become infected using a worm that scans network machines for vulnerabilities and exploits them to plant the malware.

The botnet master communicates with individual bots through a commonly used protocol, such as Internet Relay Chat (IRC), HTTP, and P2P. Bots receive commands from their master and carry out attacks as instructed without the knowledge or consent of the machine's owner. Attack types include, but are not limited to, spamming, distributed denial of service (DDoS), phishing, and identity theft. Therefore, they are considered one of the biggest threats to the internet security today. There have been multiple research efforts by industry and academia to tackle this problem [5]. These solutions generally take one of the following approaches.

Network-based solutions primarily depend on detecting the C&C channels, which leads to the identification of bot machines and server machines [6, 7].Once identified, additional actions are needed to stop their activity. While these solutions are effective for known botnets, newly developed botnets evade detection by changing their communication mechanisms and protocols. In addition to that, modern botnets use strong encryption for their C&C channels making the detection even harder [8].

Host-based solutions take a different approach by monitoring the system and detect malicious activities from the inside. The goal is to detect if the host machine has been infected by a botnet malware. Those solutions can be divided into signature-based and behavior-based. Signature-based Anti-Virus products have limited detection capability due to polymorphism techniques used in modern malware [9]. Behavior-based anti-malware solutions provide better results [10, 11], however, their effectiveness is limited when the malware has rootkit capability. Rootkits hide the malware activity on the system such that anti-malware solutions may not detect it.

Current network-based botnet-mitigation tools were used to successfully track and take down the C&C servers of certain botnets [12]. This incident will motivate botnet developers to find and use stealthier methods to hide their C&C channels. But at the end, these botnets will be used for the same kinds of attacks, such as spamming. Therefore, we believe that the detection and prevention of these types of attacks can be more effective in the fight against botnets.

3 Spamming Botnets

Email spam volume is rising in an unprecedented rate making it a major problem for the internet community today. Statistics from major email providers states that about 87% of email traffic is spam [13]. Spammers used to send large numbers of messages from their own computers. That was not efficient or reliable, however, since once spammers are discovered; their ISP takes certain actions against them. As a result, spammers have changed their tactic to sending spam from someone else's computer or by using a Botnet.

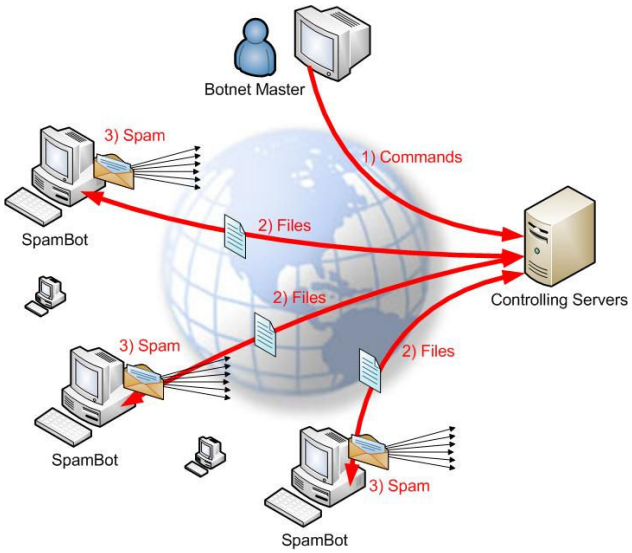

Figure 1: A Typical Spamming Botnet

Most of the spam seen today is generated by Botnets [14]. Current botnets have easy-to-use HTML-based interfaces, so they can be rented by spammers to carry out different spamming campaigns. [15]. This trend is motivated by the following reasons. Spammers can hide their

identity by using bots instead of their own machines, so they will not be held accountable for it. Second, a botnet is usually distributed among many domains, which makes tracing a single source of spam almost impossible. Third, the collective CPU and bandwidth capability allows for huge transmission of spam almost instantly. Finally, with botnets, the spamming process can be carried out in collaboration between bots performing different tasks. Some bots work as email harvesters while others work as spam generators. Some bots act as content servers while others as SMTP servers.

Spambots are pieces of malware that take the responsibility of generating and delivering spam messages. At first, spambots must be instructed to carry out a spamming campaign [16]. Each spam campaign or *workload* has at least three elements; message templates, senders list, and receivers list [17]. Spambots use these elements to generate a large volume of unique messages serving one campaign. Figure 1 illustrates the workings of a typical spamming botnet. Some bots are programmed to wait until the spamming commands are pushed to them by their controllers, while others periodically pull the spamming workload from their controlling servers or peers [16].

Recently, spamming botnets have proved their effectiveness. Researchers reported that during 2008, 85% of spam was generated by six botnets; named Mega-D, Srizbi, Storm, Rustock, Pushdo, and Cutwail [14]. Toward the end of 2008, researchers were able to locate the C&C servers of these botnets. As a result, the hosting company of these servers was taken down by its upstream ISP [12]. Immediately afterwards, the spam volume has declined but it spiked again in a short period of time [18]. This incident proves that botnet-mitigation tools cannot provide silver bullet solution for the botnet-generated spam.

4 Spam Transmission Methods

According to the SMTP protocol, typically a client or Mail User Agent (MUA) connects to a server, or Mail Transfer Agent (MTA), to transfer an email message. The MTA arranges for the delivery by forwarding (or relaying) it to another MTA. In most cases the second MTA is actually the recipient's Mail eXchange (MX) server. This process illustrated in Figure 2.

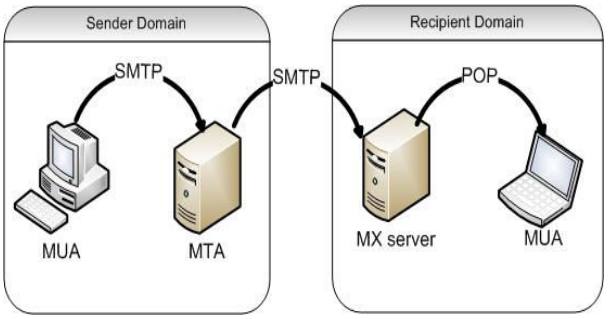

Figure 2: Legitimate Email Transmission

Botnet-generated spam does not travel this legitimate path; other methods are used to keep the spam source anonymous. In this section we illustrate these spam transmission methods and discuss their merits.

4.1 Open relay

An *open relay* is an SMTP server that accepts relay requests from any source to any destination as shown in Figure 3. Until recently, open relay was the most common method used by spammers because it was the default behavior of any SMTP server. Today's mail servers, however, come by default with open relay being disabled. Built-in checks for relay requests are developed depending on the network settings. Some ISPs use user authentication for their customers while others accept relays from IP addresses and domains that they trust.

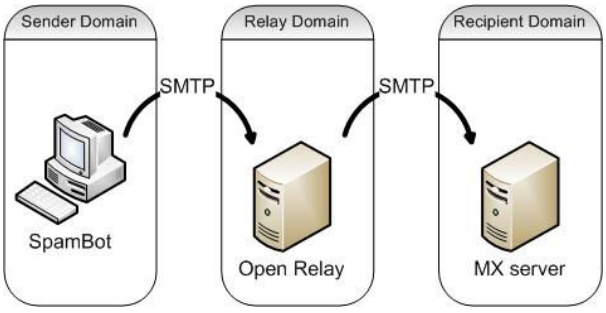

Figure 3: Open Relay

Instead of using misconfigured mail servers, which are rarely found, spambots use other bots who act as an open relay. These bots run SMTP

server on high port numbers and they may reside in the same domain or in different domains.

Relaying spam can be blocked if the spambot's network manages port 25 traffic as recommended by MAAWG [19]. With this practice, all outgoing email traffic is dropped except from legitimate mail servers. Today, an increasing number of networks are adopting these policies. As a result it has been found that open relays are rarely used in today's spamming [20].

4.2 Open Proxy

A proxy server works as an intermediary between a client and a server. It is usually used for security and privacy purposes because it changes the IP address of the client. *Open proxy* is a proxy server that allows connections to be made from any client to any server on any port. These can be legitimate proxies that have been misconfigured. Or they can be compromised machines (or bots) running a proxy service on a particular port.

Open proxies are often used by spambots in order to launder the spam traffic [21]. They receive requests from spambots and forwards spam traffic to the requested mail servers as shown in Figure 4. The origin of the spam email is thus hidden from the recipient. A study found that the top protocols used for proxying spam are HTTP and SOCKS4/ 5 [21].

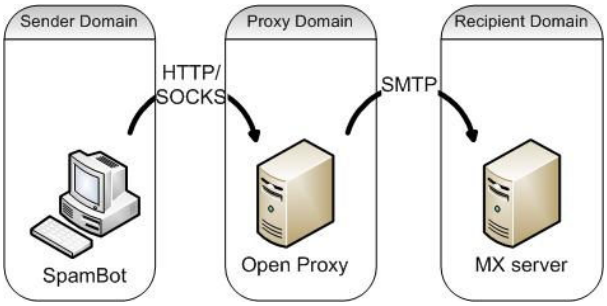

Figure 4 : Open Proxy

In order for spambots to utilize this service, they need to have IP addresses of open proxies. This can be achieved by either a network scan or by a download from the controlling servers.

Proxylock

As a special case, some spambots request the proxy to forward email packets to the MX servers of the proxy's domain. This feature is called *Proxylock* [17] as shown in Figure 5. In this case, it is the proxy's responsibility to look up the MX record of its own domain. This method achieves more than one goal; first, it makes the spam message look more legitimate because it has been relayed by a legitimate and trustworthy mail server. It also decreases the effort made by spambots to find a relay mail server. In addition to that, it transmits spam in a more distributed fashion that will disguise the original source of spam.

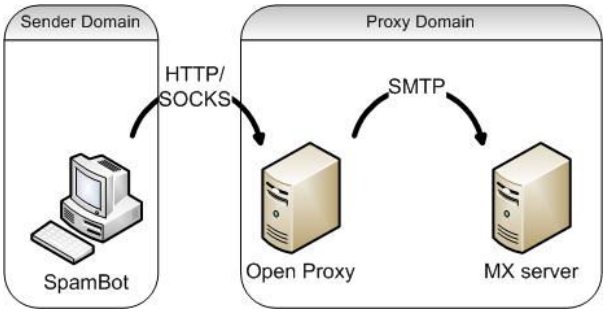

Figure 5: Proxylock

Proxylock, however, may not be useful for the following reasons. Most ISPs have separate inbound and outbound mail servers because different kinds of processing are needed in each direction. The MX record of a particular domain gives the IP addresses of the inbound mail servers. Therefore, attempts to send spam to a recipient that does not belong to that domain will fail.

Instead of using MX record, the proxy bots can use the SPF [22] record which gives the IP addresses of the authorized outbound mail servers. In addition, the bot may obtain the SMTP server settings from the email client of the infected machine. But even if the bot was able to find the proper mail server, major ISPs employ policies such as rate limiting and user authentication that prevent such activity from happening.

4.3 Direct-to-MX

Botnet generated spam can be delivered directly to the MX server of the recipient's domain as shown in Figure 6. For legitimate emails and according to SMTP protocol, direct delivery is done by the MTAs, not by the end users or their MUAs. MTAs query the DNS for the MX record of the recipient's domain and then deliver the message to that server.

In order for a spambot to do that; it must obtain the IP address of these servers for each email address on it list. It can query the DNS for MX record of all the recipients' domains, but this slows down the spamming process. A more advanced way is to download the MX records along with email lists from the botnet servers or peers. This needs extra management efforts for maintaining and updating these lists by botnet controllers.

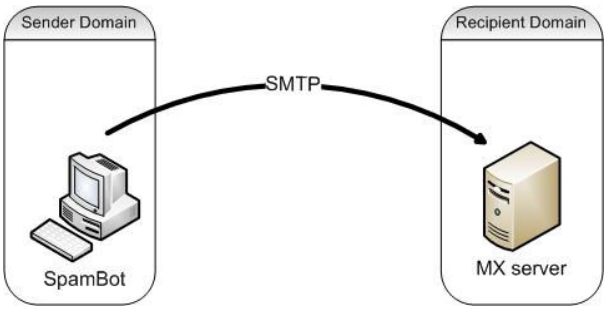

Figure 6: Direct-to-MX

Direct delivery is favored by spamming botnets because it reduces the chance of filtering out spam messages by intermediate relays. On the other hand, it imposes the risk of blacklisting the IP address of the spambot. Blacklisting usually takes a long time after reporting the abuse, leaving the spambot active in the meanwhile.

One important limitation for this method is realized when the ISP of the spambot manages port 25 traffic [19]. This usually causes all outgoing email traffic to be dropped except from specified mail servers. This will prevent the spambots from delivering any email messages to the outside domains.

5 Countermeasures

Measures against a successful delivery of spam can be taken along the message transmission path. The path from the source network to the destination network includes routers, and mail servers. Measures that are taken at edge router are different from the ones taken at mail servers due to the type of information that is available at each one. At routers, network-level information is only available, while at mail servers, application-level information is also available. Therefore, we divide the methods discussed here by the location where they can be implemented.

5.1 Edge Router

Networks administrators usually adapt certain policies for managing the upstream and downstream network traffic passing through the edge routers. Certain policies can significantly reduce the chance of receiving or sourcing spam traffic generated by botnets. In this section we survey methods for controlling incoming and outgoing spam at the network level. Table 1 gives a summary of these methods.

5.1.1 Incoming Spam Control

Cook et al. developed a system to block spam traffic at the router level [23]. Traffic from a certain IP address is blocked if the IP address has previously sent a spam message. It maintains a local blacklist at the network edge routers. An IP address remains blocked for a certain period of time determined by network operators. This method relies on spam filtering at the mail server level so at least one spam message has to pass before others get blocked. With this method legitimate email messages from the spambots machine can be blocked.

Argawal et al. developed a system that studies incoming email traffic and identifies bulk spam streams [24]. They assume that such streams contain a large number of similar messages. A bulk email stream is identified by utilizing a caching mechanism on the routers. Then for each stream, the system determines if it is spam or not using a Bayesian classifier. Once spam streams are identified, network operators can impose rate limiting on them. This system assumes that spam comes in bulk, while with spamming botnets it might not be the case. In addition, finding similarity between spam messages is challenged because current spambots uses templates and able to produce unique looking messages carrying the same contents.

A number of studies has been conducted to analyze spam traffic and to find its distinguishing characteristics. Beverly and Sollins studied TCP flow features and found that spam traffic has distinguishable Round Trip Times (RTTs), maximum idle times, and FIN packet counts [25]. They designed **SpamFlow**, a machine learning classifier trained on email connections to an open relay MTA. Their study did not include spam directly delivered to mail servers. That may add different selective features to the classifier.

5.1.2 Outgoing Spam Control

Some ISPs adopt MAAWG [19] recommendations of **managing port 25 traffic**. It states that egress routers should only allow mail traffic from certain mail servers and drop all other. This can significantly reduce the chance of sourcing spam. Unfortunately that cannot be applied in all network settings. Therefore, spambots residing in such networks can send out spam unless other measures are used.

Romana et al. analyzed the entropy values of DNS query traffic in a university campus [26]. The study revealed that spambots can be discovered because they frequently issue DNS queries for MX records. They also found that the queried mail servers were randomly distributed and do not belong to a certain domain. With this method, spambots residing inside the network could be identified and blocked. Currently, some spambots do not issue DNS queries because they maintain a list of target MX records, causing this method to fail.

As mentioned in the previous section, spambots may use proxies to hide their spam traffic. Xie et al. developed a technique, called **DBSpam**, to detect and interrupt this kind of activity [27]. By Monitoring SMTP traffic on edge routers, a timely correlation between SMTP replies and proxy traffic was revealed. This correlation is used in identifying the spam network flows. After detection, DBSpam can either throttle or block the spam traffic.

Table 1: Spam control at the network-level

Method	Direction	Effect
Cook	in	Block email traffic from locally-blacklisted sources
Argawal	In	Detect bulk spam traffic
SpamFlow	In	Detect spam TCP flows
Manage Port 25	Out	Drop email traffic except from predefined servers
Romana	Out	Detect spambots DNS MX queries
DBSpam	In/Out	Block/Throttle proxying

5.2 Mail Server

Most ISPs are currently adapting email best practices outlined by MAAWG [28]. One of these practices is to separate mail servers by function into inbound and outbound. This is important because different kind of processing is needed for each direction. Therefore, spam control at the server level can be implemented by MTAs and MX servers. The following are some methods for controlling spam where the goal is to only accept legitimate email requests and deny all others.

5.2.1 MTA Spam Control

Most MTAs today are configured to **reject open relay** requests. They only accept connections from machines inside the network. This measure prevents relay attempts by spambots residing outside the network but it allows if for insiders.

Some ISPs enforce **email forwarding best practices** as recommended by MAAWG [28]. They require a different port number usage (other than 25) and user authentication. This measure prevents spambots residing inside the network from relaying spam through them. But if the spambot knows the port number and the user's credentials of the infected machine, they can accomplish that successfully.

Another way to stop spam at the MTAs is to employ an **SMTP Transaction Delay** [29]. These

delays are imposed selectively on suspicious forwarding attempts. Identifying suspicious attempts employ other heuristics such as DNS checks or SMTP protocol compatibility checks. Imposing this kind of delay can reduce the total number of spam delivered, but it makes the server vulnerable to denial of service for busy networks.

5.2.2 Incoming Spam Control

Spam control at the MX server level can employ many heuristics depending on the message header and content. SpamAssasin [30] is one example. We are only interested in methods that do not rely on the message content because it is usually expensive to do. The goal of the methods discussed here is to detect transmission attempts by spambots residing outside the mail server's network.

Mail servers can check if the sending IP is an **authorized mail server** in the domain where it claims to be. This method depends primarily on DNS specialized records. Three different systems have been designed for this purpose; Domain Keys Identified Mail (DKIM) [31], Sender ID [32], and Sender Policy Framework (SPF) [22]. This measure block messages from spambots residing in domains that use this service. But it does not do that if the spambot's domain is not having such records.

Mail servers can use the reputation of the mail source in deciding whether to accept or reject a request. In recent years, the reputation is decided by **blacklists** such as Spamhaus Block List (SBL) [33] for sources of spam and SORBS [34] for open relays and open proxies. This measure block messages from spambots that has been blacklisted only. A study found that 35% of spam comes from sources not listed in any blacklist [2]. Behavioral blacklisting can be used instead, as proposed by Ramachandran et al [35].

Greylisting [36] is related to blacklists and whitelists, but it can be powerful in delaying and refusing spam. With this method all delivery attempts are refused until the sender tries again. This can be useful because spambots' implementation of SMTP may not comply completely with the protocol requirements.

Another method similar to greylisting is **SMTP session abort** [37]. Instead of refusing the first delivery attempt, mail server abort the connection after it obtained the message header and body. Then if the message is not resent, it is registered as suspicious. Saving this information is used for future identification of spam messages.

Table 2: Spam control at the server-level

Method	Direction	Effect
Reject open relays	In/Out	Block open relay attempts
Forwarding best practices	Out	Drop email from unauthorized users
SMTP delay	In	Delay spam and reduce its volume
Source checking	In	Drop email from untrusted servers
Greylisting	In	Refuse delivery attempts by untrusted sources
SMTP abort	In	Refuse delivery attempts from known suspicious sources

6 Conclusions and Future Work

In this paper we tried to shed light on the transmission methods used by current spamming botnets. We believe that measures at the network level can be very effective in neutralizing spambots. The first case is when spambots reside inside a network. Spam relay and delivery attempts can be prevented when email traffic is managed according to MAAWG recommendation. In cases where this cannot be adopted, monitoring outgoing email traffic can give an indication of spamming activities:

- Romana was able to detect spambots because they issue frequent DNS queries

[26]. Advanced botnets, however, maintain specialized databases for MX records, thus DNS is not needed or used.

- DBSpam can detect and block proxy channels used for spam laundering[27]. Regular spam traffic, however, is not detected.

The second case is when spambots reside outside of the network. According to the mentioned studies, spam traffic can be distinguished from legitimate mail traffic:

- Argawal assumed that spam comes in bulk, and was able to distinguish it from legitimate bulk using spam content filtering [24]. Botnet-generated spam is generated by templates making each message unique, thus defining a certain bulk becomes challenging.
- SpamFlow can find spam distinguishing TCP features using machine learning [25]. It is important, however, to use features that spambots cannot adapt to.

Network-level spam control will be adopted at routers, so it is imperative to consider the performance of these tools and how it affects the router speed. The methods presented in this paper do not include such analysis making it an opportunity for future work.

Spam control on the server-level provides an additional layer of protection. Spambots residing inside the network cannot relay spam though the network mail servers if user authentication is adopted. Unfortunately, some network settings does not allow for the adoption of these policies, so other methods are needed:

- Authorized mail server checking can be very effective if authentication systems are adopted widely. For the time being, this is not the case.
- Blacklists checking can be very effective too, but there is a delay between discovering a spambot and getting it blacklisted. Within this timeframe, spambots can generate and transmit large volumes of spam.
- SMTP delays can deter spammers from relaying spam because that slows down the transmission of high volumes of messages. With botnets, imposing delays should not impact the process because large numbers of bots are utilized.
- Greylisting and similar methods can drop spam from non-RFC compliant spambots, but that can easily be fixed by botnet developers.

In summary, utilizing botnets for spamming brings out new challenges. Anti-spam solutions need to be adjusted taking into considerations the distributed and anonymous nature of botnet-generated spam. Fighting botnets is challenging and need to be taken at many levels. In addition, collaboration is needed between different entities, such ISPs, network administrators, and end users.

References

[1] P. Buxbaum, "Battling Botnets," *Military Information Technology (MIT),* vol. 12, 2008.

[2] A. Ramachandran and N. Feamster, "Understanding the network-level behavior of spammers," *SIGCOMM Comput. Commun. Rev.,* vol. 36, pp. 291-302, 2006.

[3] Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten, and I. Osipkov, "Spamming botnets: signatures and characteristics," in *Proceedings of the ACM SIGCOMM 2008 conference on Data communication* Seattle, WA, USA: ACM, 2008.

[4] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "A multifaceted approach to understanding the botnet phenomenon," in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement* Rio de Janeriro, Brazil: ACM, 2006.

[5] Z. Zhaosheng, L. Guohan, C. Yan, Z. J. Fu, P. Roberts, and H. Keesook, "Botnet Research Survey," in *Computer Software and Applications, 2008. COMPSAC '08. 32nd Annual IEEE International*, 2008, pp. 967-972.

[6] G. Gu, J. Zhang, and W. Lee, "BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic," in *15th Annual Network & Distributed System Security Symposium*, San Diego, CA, 2008.

[7] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "BotMiner: clustering analysis of network traffic for protocol- and structure-independent botnet detection," in *Proceedings of the 17th conference on Security symposium* San Jose, CA: USENIX Association, 2008.

[8] R. Hund, M. Hamann, and T. Holz, "Towards Next-Generation Botnets," in *Computer Network Defense, 2008. EC2ND 2008. European Conference on*, 2008, pp. 33-40.

[9] K. J. Higgins, "Study: Antivirus Software Catches About Half Of Malware, Misses 15 Percent Altogether " 2009, http://www.darkreading.com/

[10] M. D. Preda, M. Christodorescu, S. Jha, and S. Debray, "A semantics-based approach to malware detection," in *Proceedings of the 34th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages* Nice, France: ACM, 2007.

[11] E. Stinson and J. C. Mitchell, "Characterizing Bots' Remote Control Behavior," in *Proceedings of the 4th international conference on Detection of Intrusions and Malware, and Vulnerability Assessment* Lucerne, Switzerland: Springer-Verlag, 2007.

[12] B. Kerbs, "Host of Internet Spam Groups Is Cut Off," in *The Washington Post*, 2008.

[13] Messaging Anti-Abuse Working Group (MAAWG), "Email Metrics Report," 2008, http://www.maawg.org/

[14] J. Carr, "TRACE: Six botnets generate 85 percent of spam," in *SC Magazine*, 2008.

[15] J. Stewart, "Spam Botnets to Watch in 2009," 2009, http://www.secureworks.com/

[16] C. Kreibich, C. Kanich, K. Levchenko, B. Enright, G. M. Voelker, V. Paxson, and S. Savage, "On the spam campaign trail," in *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats* San Francisco, California: USENIX Association, 2008.

[17] H. Stern, "A Survey of Modern Spam Tools," in *Proceedings of the Fifth Conference on Email and Anti-Spam (CEAS)* Mountain View, CA, 2008.

[18] TRACE at Marshal, "Spam Statistics," 2009, http://www.marshal.com/trace/spam_statistics.asp

[19] Messaging Anti-Abuse Working Group (MAAWG), "Managing Port 25 for Residential or Dynamic IP Space," 2005, http://www.maawg.org/

[20] P. H. Calais, D. E. V. Pires, D. O. Guedes, W. M. Jr., C. Hoepers, and K. Steding-Jessen, "A Campaign-based Characterization of Spamming Strategies," in *Proceedings of the Fifth Conference on Email and Anti-Spam (CEAS)* Mountain View, CA, 2008.

[21] K. Steding-Jessen, N. L. Vijaykumar, and A. Montes, "Using Low-Interaction Honeypots to Study the Abuse of Open Proxies to Send Spam," *INFOCOMP - Journal of Computer Science,* vol. 7, pp. 44-52, March 2008.

[22] "Sender Policy Framework (SPF)," http://www.openspf.org/

[23] D. Cook, J. Hartnett, K. Manderson, and J. Scanlan, "Catching spam before it arrives: domain specific dynamic blacklists," in *Proceedings of the 2006 Australasian workshops on Grid computing and e-research - Volume 54* Hobart, Tasmania, Australia: Australian Computer Society, Inc., 2006.

[24] B. Argawal, N. Kumar, and M. Molle, "Controlling spam Emails at the routers," in *Communications, 2005. ICC 2005. 2005 IEEE International Conference on*, 2005, pp. 1588-1592 Vol. 3.

[25] R. Beverly and K. Sollins, "Exploiting Transport-Level Characteristics of Spam," in *Proceedings of the Fifth Conference on Email and Anti-Spam (CEAS)* Mountain View, CA, 2008.

[26] D. A. L. Romaa, S. Kubota, K. Sugitani, and Y. Musashi, "DNS Based Spam Bots Detection in a University," in *Intelligent Networks*

and Intelligent Systems, 2008. ICINIS '08. First International Conference on, 2008, pp. 205-208.

[27] M. Xie, H. Yin, and H. Wang, "An effective defense against email spam laundering," in *Proceedings of the 13th ACM conference on Computer and communications security*, Alexandria, Virginia, USA, 2006.

[28] Messaging Anti-Abuse Working Group (MAAWG), "Email Forwarding Best Practices," 2008, http://www.maawg.org/

[29] T. Slettnes, "Spam Filtering for Mail Exchangers."

[30] "SpamAssasin," http://spamassassin.apache.org/

[31] "Domain Keys Identified Mail (DKIM)," http://www.dkim.org/

[32] Microsoft, "Sender ID," 2008, http://www.microsoft.com/

[33] The SPAMHAUS Project, "The Spamhaus Block List," 2008, http://www.spamhaus.org/

[34] "Spam and Open Relay Blocking System (SORBS)," http://www.au.sorbs.net/

[35] A. Ramachandran, N. Feamster, and S. Vempala, "Filtering spam with behavioral blacklisting," in *Proceedings of the 14th ACM conference on Computer and communications security* Alexandria, Virginia, USA: ACM, 2007.

[36] "Greylisting," http://www.greylisting.org/

[37] N. Yamai, K. Okayama, T. Seike, K. Kawano, M. Nakamura, and S. Maruyama, "An Anti-Spam Method with SMTP Session Abort," in *MIT Spam Conference*, 2008.